**AIRLINES, AIRPORT AND ROUTES   
Report By : Ali A Nathani  
Instructor: Prof. Anasuya Bhima**

* Introduction & Project Goals:

The goal of the project is to understand big data in more depth and have practical experience of working with different tools in Hadoop to handle Big Data. The practical steps are in 2 phases as below:

**PHASE 1 (LOADING THE DATASETS INTO THE LINUX SYSTEM AND THEN MOVING IT TO THE HDFS):  
1. Create a Folder on the Linux /Sandbox system named ‘project’  
2. Move all the data sets files into the folder using FileZilla  
3. Create a /user/project folder on the HDFS system  
4. Copy or move files into the new folder on the HDFS system**

**Data source downloaded from** <https://openflights.org/data.html>  
3 files downloaded, airports.dat, routes.dat and airlines.dat

**The above 3 datasets have 67663 routes between 3321 airports on 548 airlines** spanning the globe as of June 2014  
Attributes are as follows:

**Airlines.dat**

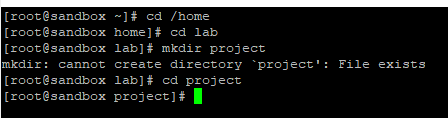
|  |  |
| --- | --- |
| **Airport ID** | Unique OpenFlights identifier for this airport. |
| **Name** | Name of airport. May or may not contain the **City** name. |
| **City** | Main city served by airport. May be spelled differently from **Name**. |
| **Country** | Country or territory where airport is located. See [countries.dat](https://raw.githubusercontent.com/jpatokal/openflights/master/data/countries.dat) to cross-reference to ISO 3166-1 codes. |
| **IATA** | 3-letter IATA code. Null if not assigned/unknown. |
| **ICAO** | 4-letter ICAO code. Null if not assigned. |
| **Latitude** | Decimal degrees, usually to six significant digits. Negative is South, positive is North. |
| **Longitude** | Decimal degrees, usually to six significant digits. Negative is West, positive is East. |
| **Altitude** | In feet. |
| **Timezone** | Hours offset from UTC. Fractional hours are expressed as decimals, eg. India is 5.5. |
| **DST** | Daylight savings time. One of E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None) or U (Unknown). *See also:*[*Help: Time*](https://openflights.org/help/time.html) |
| **Tz database time zone** | Timezone in ["tz" (Olson) format](http://en.wikipedia.org/wiki/Tz_database), eg. "America/Los\_Angeles". |
| **Type** | Type of the airport. Value "airport" for air terminals, "station" for train stations, "port" for ferry terminals and "unknown" if not known. *In airports.csv, only type=airport is included.* |
| **Source** | Source of this data. "OurAirports" for data sourced from [OurAirports](http://ourairports.com/data/), "Legacy" for old data not matched to OurAirports (mostly DAFIF), "User" for unverified user contributions. *In airports.csv, only source=OurAirports is included.* |

**Airports.dat**.

|  |  |
| --- | --- |
| **Airline ID** | Unique OpenFlights identifier for this airline. |
| **Name** | Name of the airline. |
| **Alias** | Alias of the airline. For example, All Nippon Airways is commonly known as "ANA". |
| **IATA** | 2-letter IATA code, if available. |
| **ICAO** | 3-letter ICAO code, if available. |
| **Callsign** | Airline callsign. |
| **Country** | Country or territory where airline is incorporated. |
| **Active** | "Y" if the airline is or has until recently been operational, "N" if it is defunct. This field is *not* reliable: in particular, major airlines that stopped flying long ago, but have not had their IATA code reassigned (eg. Ansett/AN), will incorrectly show as "Y". |

**Routes.dat**

|  |  |
| --- | --- |
| **Airline** | 2-letter (IATA) or 3-letter (ICAO) code of the airline. |
| **Airline ID** | Unique OpenFlights identifier for airline (see [Airline](https://openflights.org/data.html#airline)). |
| **Source airport** | 3-letter (IATA) or 4-letter (ICAO) code of the source airport. |
| **Source airport ID** | Unique OpenFlights identifier for source airport (see [Airport](https://openflights.org/data.html#airport)) |
| **Destination airport** | 3-letter (IATA) or 4-letter (ICAO) code of the destination airport. |
| **Destination airport ID** | Unique OpenFlights identifier for destination airport (see [Airport](https://openflights.org/data.html#airport)) |
| **Codeshare** | "Y" if this flight is a codeshare (that is, not operated by *Airline*, but another carrier), empty otherwise. |
| **Stops** | Number of stops on this flight ("0" for direct) |
| **Equipment** | 3-letter codes for plane type(s) generally used on this flight, separated by spaces |

#Step 1. Preparing the folders for importing data  
#In HortonWorks and connected via Putty.exe within the Linux box (Sandbox), use following code to  
open the home directory and make a folder within the lab folder

cd /home  
cd lab   
mkdir project

#Step 2. Using FileZilla Site Manager, transfer the .dat files to the project folder

#Step 3. make an hdfs project folder (if doesn't already exist) using the following code from within the Linux box (Sandbox)  
hdfs dfs -mkdir /user/project

#Step 4. transfer the files from the local Linux box to HDFS  
#first open the project folder on local box  
cd project

#then transfer all files (specify file names instead of \* if you want to do 1 at a time) to the HDFS box

hdfs dfs -put \* /user/project

**PHASE 2 (USING THE PIG OR HIVE TOOLS TO SOLVE THE FOLLOWING QUERIES:)** **#Q1 FIRST 5 AIRLINES IN ALPHABETICAL ORDER AND TOP 5 COUNTRIES WITH MOST AIRLINES  
#Q2 FINDING THE NUMBER OF AIRPORTS IN EACH COUNTRY - TOP 5 COUNTRIES  
#Q3 FINDING THE DISTINCT ROUTES BETWEEN SOURCE AND DESTINATION AIRPORTS, Submit the first five rows.  
#Q4 FIND THE DISTANCES BETWEEN LATITUDE AND LONGITUDE BETWEEN ROUTES, SORT DISTANCES AND OUTPUT TOP 5, ALSO without sorting find the top 5**

**PHASE 2 -**  **#Q1 FIRST 5 AIRLINES IN ALPHABETICAL ORDER  
Tool used: Pig, Data file used: airlines.dat**

|  |
| --- |
| a = load '/user/project/airlines.dat' USING PigStorage(',') AS (airline\_id:long, name:chararray, alias:chararray, IATA:chararray, ICAO:chararray, callsign:chararray, country:chararray, active:chararray);  #Run the following code to list the distinct airlines:  b = foreach a generate airline\_id, LOWER(name) as name;  c = distinct b;  d = limit c 5;  STORE d into '/user/project/distinct\_airlines' using PigStorage('\t') ;  #to check the temp file name created for the output  fs -ls /user/project/distinct\_airlines;  #to get the output printed on screen instead of using the Dump command which also shows all codes  fs -cat /user/project/distinct\_airlines/part-v002-o000-r-00000; |
| **Explanation:** The above code gives an output of the first 5 airlines in the airlines.dat dataset.  **FINDINGS: The first 5 airlines names, alphabetically, actually start with numbers. There are several small airlines that are uncommon.** **OUTPUT** |
| **Check top 5 countries with most airlines:** |

**Explanation:** The United States has the most number of airlines, 1099, followed by Mexico, United Kingdom, Canada and Russia.

**PHASE 2 - #Q2 FINDING THE NUMBER OF AIRPORTS IN EACH COUNTRY - TOP 5 COUNTRIES  
Tool used: Pig, Data file used: airports.dat**

|  |
| --- |
| a = load '/user/project/airports.dat' using PigStorage(',') AS (airport\_id: chararray, name:chararray, city:chararray, country:chararray, IATA:chararray, ICAO:chararray, lat:float, long:float, altitude:long, timezone:float, DST:chararray, Tzdata:chararray, type:chararray, source:chararray);  b = group a by country;  c = foreach b generate group as country, COUNT(a) as cnt;  d = order c by cnt desc;  e = limit d 5;  STORE e into '/user/project/top\_airports' using PigStorage('\t') ;  #to check the temp file name created for the output  fs -ls /user/project/top\_airports;  #to get the output printed on screen instead of using the Dump command which also shows all codes  #FIRST 5 AIRLINES IN ALPHABETICAL ORDER  fs -cat /user/project/top\_airports/part-v005-o000-r-00000; |
| **Explanation:** The above code first groups and counts the countries as a new variable ‘cnt’. This is sorted in descending order and then the top 5 are stored into the external file. **FINDINGS : The United States has the largest number of active Airports, followed by Canada, Australia, Germany and Russia**  **OUTPUT:** | |

**PHASE 2 - #Q3 FINDING THE DISTINCT ROUTES BETWEEN SOURCE AND DESTINATION AIRPORTS, Use the first five rows. Get the names of the Destination airports for these routes.  
Tool used: Pig and then Hive, Data file used: routes.dat**

|  |
| --- |
| a = load '/user/project/routes.dat' using PigStorage(',') AS (airline:chararray, ID:long, source:chararray, sid:long, dest:chararray, did:chararray, codeshare:chararray, direct:int, equip:long);  b = foreach a generate source, sid, dest, did;  c = distinct b;  d = limit c 5;  STORE d into '/user/project/distinct\_routes' using PigStorage('\t') ;  #go back to linux box and launch hive  #create the routes table in hive after creating a database  create database airlines;  use airlines;  create table airports (  airport\_id string,  name string,  city string,  country string,  IATA string,  ICAO string,  lat string,  lon string,  alt string,  timezone string,  DST string,  Tz string,  type string,  Source string)  row format delimited  fields terminated by ',';  load data inpath '/user/project/airports.dat' overwrite into table airlines.airports;  create table dist\_routes (  source string,  sid bigint,  dest string,  did bigint)  row format delimited  fields terminated by '\t';  load data inpath '/user/project/distinct\_routes/part-v002-o000-r-00000' overwrite into table airlines.dist\_routes;    select \* from dist\_routes;    select a.source,a.sid, a.dest,a.did, b.name, b.city, b.country  from dist\_routes a left outer join airports b  on a.did = b.airport\_id; |

**Explanation:** The above code was first run in Pig to get the distinct routes. The output was then saved and transferred into the Tables created in Hive using the above mentioned code. Once the data was loaded in the relevant tables, the Query was run to see the top 5 airports names of the destinations of the distinct routes.  
**FINDINGS : The Source of all the top 5 disctinct routes was the same, AAE, also in Algeria, however, France was the most common in the top 5 destination list from AAE.**

**PHASE 2- #Q4 DETERMINE THE DISTANCES TRAVELLED ON EACH ROUTE USING distance in kilometres using the simple Euclidian formula:**

**distance = SQRT((lat2 - lat1) \* (lat2 - lat1) + (lon2 - lon1) \* (lon2 - lon1)) \* 111,**

**SORT DESCENDING AND DISPLAY THE TOP 5 WITH LONGEST DISTANCES  
Tool used: Pig**

|  |
| --- |
| a = load '/user/project/routes.dat' using PigStorage (',') AS (airline:chararray, ID:long, source:chararray, sid:chararray, dest:chararray, did:chararray, codeshare:chararray, direct:int , equip:long);  b = load '/user/project/airports.dat' using PigStorage(',') AS (airport\_id:chararray , name:chararray , city:chararray , country:chararray , IATA:chararray , ICAO:chararray , lat:float , lon:float , altitude:long , timezone:float , DST:chararray , Tzdata:chararray , type:chararray , src:chararray);  c = join a by sid, b by airport\_id;  d = foreach c generate sid , source , lat as lats , lon as lons , dest , did;  e = join d by did , b by airport\_id;  f = foreach e generate sid , source , lats , lons , did , dest , lat as latd , lon as lond;  g = foreach f generate sid , source , did , dest , SQRT((latd - lats) \* (latd- lats) + (lond - lons) \* (lond - lons)) \* 111 as distance;  h = order g by distance desc;  i = limit h 5;  dump i; |
| **OUTPUT:**  **Explanation:** The top 5 routes seem to be mostly for the same destination and above the average range of 39,500kms.  **Without sorting in Descending order, what are the top 5 routes**  j = foreach f generate sid , source , did , dest , SQRT((latd - lats) \* (latd- lats) + (lond - lons) \* (lond - lons)) \* 111 as distance2;  k = order j by distance2;  l = limit k 5;  dump l; |